ML PROJECT: **CNN-Based Speech Command Recognition**

One-Page Project Summary

Problem Statement
The increasing presence of voice-controlled devices demands accurate and efficient command recognition from short audio clips. This project focuses on developing a system that can classify simple speech commands (like "up", "down", "go", etc.) in real-time audio, enabling robust human-computer interaction for applications such as virtual assistants and smart devices.

Approach
To address this problem, the project leverages deep learning using a Convolutional Neural Network (CNN) architecture tailored for processing Mel-Frequency Cepstral Coefficient (MFCC) features extracted from short (1-second) WAV audio samples. The dataset used is Google's Mini Speech Commands, containing 8 command categories with balanced audio samples for each.

Implementation Overview

- **Data Preparation**: The Mini Speech Commands dataset is downloaded, unpacked, and organized into labeled folders. Each audio file is preprocessed and converted to MFCC features with shape (98, 40), ensuring compatibility with the CNN input.
- **Model Architecture**: A CNN was constructed with two convolutional layers (using large receptive fields), max pooling, and dropout for regularization. The model flattens the output and passes it through dense layers ending with a softmax classifier.
- **Training**: The audio data is split into train, validation, and test sets (70/10/20 split). The model is trained for up to 30 epochs using Adam optimizer and early stopping to prevent overfitting, with hyperparameters (learning rate, batch size, dropout) optimized for best validation accuracy.
- **Evaluation**: Test accuracy, confusion matrix, and detailed precision/recall metrics are reported after training. Batch prediction and per-sample confidence scoring help interpret the model's correctness.
- **Deployment Ready**: The trained model is exported to both HDF5 and native Keras formats and can be used for inference with new audio files and within production systems.

Results and Conclusions

- The best model consistently achieves **test accuracy around 84%–87%** across the 8 command classes, with per-class precision/recall mostly above 80%.
- The CNN approach substantially outperforms simpler baselines by utilizing spatial audio features.
- Deployment scripts and functions allow individual and batch prediction, with confidence scores for each class.

Challenges and Lessons Learned

- **GPU Availability**: Training is much faster on GPUs; lack of runtime GPU can be a bottleneck.
- **Legacy Formats**: Modern Keras versions recommend .keras instead of .h5 for saving models.

- **Audio Quality and Uniformity**: Ensuring input audio matches expected sample rate and duration is critical for reliability.
- **Class Imbalance/Confusion**: Some commands (e.g., "no"/"go"/"down") can be acoustically similar, resulting in confusion even for well-trained models.
- **Generalization**: Model is sensitive to environmental noise; data augmentation or noise layers could further help robustness.

This project demonstrates how deep learning with CNNs can effectively solve speech command recognition tasks, and provides a ready-to-use pipeline for further experimentation or real-world deployment.

End of One-Page Write-up